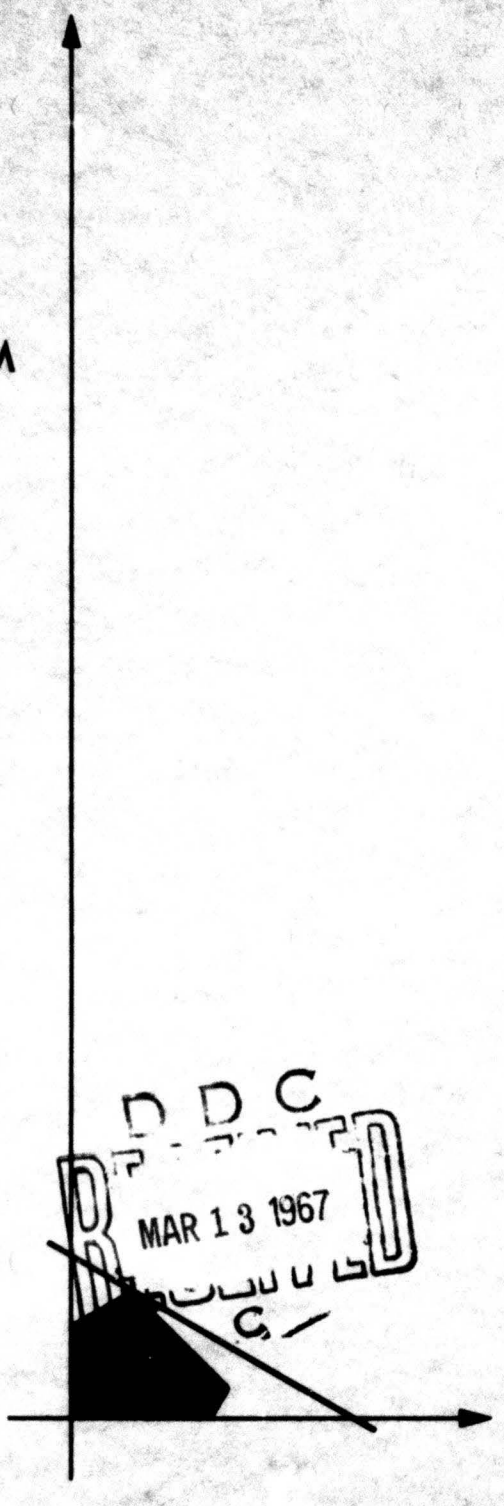
ORC 66-39
NOVEMBER 1966

A NEW FOUNDATION FOR A SIMPLIFIED PRIMAL INTEGER PROGRAMMING ALGORITHM

by

Fred Glover



OPERATIONS RESEARCH CENTER

COLLEGE OF ENGINEERING

UNIVERSITY OF CALIFORNIA - BERKELEY

A NEW FOUNDATION FOR A SIMPLIFIED PRIMAL INTEGER PROGRAMMING ALGORITHM

by

Fred Glover
Operations Research Center
University of California, Berkeley

November 1966 ORC 66-39

This research was supported in part by the National Science Foundation, Grant GP-4593, and the Office of Naval Research under Contract Nonr-222(83), with the University of California. Reproduction in whole or in part is permitted for any purpose of the United States Government.

A NEW FOUNDATION FOR A SIMPLIFIED PRIMAL INTEGER PROGRAMMING ALGORITHM*

by

Fred Glover

1. INTRODUCTION

The purpose of this paper is to show that the conceptual foundations and presentation of R. D. Young's Simplified Primal Integer Programming Algorithm [8] can be simplified further on the basis of a few fundamental algebraic relations. These relations derive from the approach underlying the author's Pseudo Primal-Dual Integer Programming Algorithm [3] which contributed to the basis for [8] and, in their present development, provide new choice rules that yield a finite primal method. In addition, a criterion of optimality is introduced that generally permits the algorithm to be terminated before the customary optimality conditions are manifested.

2. DESCRIPTION OF THE PROBLEM

We represent the ordinary linear programming problem P1 as that of maximizing in nonnegative variables

$$x_0 = a_{00} + \sum_{j=1}^{n} a_{0j}(-t_j)$$

subject to

$$x_i = a_{i0} + \sum_{j=1}^{n} a_{ij}(-t_j) \quad , \; i = 1, \ldots, m \; , \tag{1}$$

$$x_{m+j} = -(-t_j) \quad , \; j = 1, \ldots, n \; ,$$

*The author is deeply indebted to Professor Richard D. Young for many stimulating discussions that have contributed to the development of this paper.

or in matrix form to maximize x_0 subject to

$$X = A^0T^0 \ , \ A^0 = (A_0, A_1, \ldots, A_n)$$

$$X = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_{m+n} \end{bmatrix} \ , \ T = \begin{bmatrix} 1 \\ -t_1 \\ -t_2 \\ \cdot \\ \cdot \\ \cdot \\ -t_n \end{bmatrix} .$$

The matrix A^0 is *dual feasible* if $a_{0j} \geq 0$ for $j = 1, \ldots, n$, and *primal feasible* if $a_{i0} \geq 0$ for $i = 1, \ldots, m + n$. As is well known, an optimal solution to P1 is immediately given by $X = A_0$ when both primal and dual feasibility hold.

The pure integer programming problem P2, which provides the chief focus of this paper, is the same as P1 except that the components of X are additionally required to be integers. Following the lead of R. D. Young [6, 7], we will specify a method for solving P2 that yields a nonnegative (primal feasible) integer X and a nondecreasing value of x_0 at each stage of the solution process.† To provide a foundation for this method, we review the version of the simplex algorithm that exhibits the same characteristics in solving P1 except that the successive X vectors may not be integer.

†The value of such an approach is at least three fold. First, it is possible to begin with a known feasible integer solution and obtain progressively better ones. Second, one may discontinue the process of solving P2 at any stage and still have a workable, if not optimal, solution. Third, the method typically provides a range of feasible integer solutions instead of single best one, which may be useful in certain applications. (These features may of course be of little advantage in solving problems that are extremely resistant to the method, unless such problems are correspondingly difficult for other algorithms.)

3. THE PRIMAL SIMPLEX ALGORITHM (PSA)

Beginning with A^0 primal feasible, the primal simplex method for solving P1 determines a sequence of representations for X :

$$X = A^0T^0 = A^1T^1 = A^2T^2 = \ldots = A^kT^k ,$$

$$T^h = (1, -t_1^h, -t_2^h, \ldots, -t_n^h) , \quad h = 0, 1, \ldots, k$$

where t_j^h $(j = 1, \ldots, n)$ is nonnegative, $a_{00}^h \leq a_{00}^{h+1}$, and A^h is primal feasible for each h . When P1 is bounded for optimality, the matrix A^k (for finite k) is also dual feasible.

For simplicity, we will let A and T denote any matrix A^h and vector T^h , and let $\bar{A}$ and $\bar{T}$ denote the corresponding A^{h+1} and T^{h+1} . Then the precise rules of the PSA are as follows:

1. If $a_{0j} \geq 0$ for all $j \geq 1$, then $X = A_0$ is an optimal solution. Otherwise, select $s \geq 1$ such that $a_{0s} < 0$.
2. If $a_{is} \leq 0$ for all $i \geq 1$, P1 has an unbounded optimum. Otherwise, compute $\theta_s = \underset{a_{is}>0}{\text{Min}} \ (a_{i0}/a_{is})$.
3. Select v such that $a_{vs} > 0$ and $a_{v0}/a_{vs} = \theta_s$.[+]
4. Determine $\bar{A}$ by the rules:

$$\bar{A}_s = -A_s/a_{vs}$$

$$\bar{A}_j = A_j - A_s(a_{vj}/a_{vs}) \quad \text{if} \quad j \neq s .$$

5. Let $\bar{t}_s \equiv x_v$ and $\bar{t}_j \equiv t_j$ for $j \neq s$. Designate $\bar{A}$ and $\bar{T}$ to be the current A matrix and T vector, and return to instruction 1.

[+] We do not concern ourselves with tiebreaking rules in the choice of v , since those that assure finiteness for the PSA have little bearing on finiteness for a primal integer method.

Because the PSA is quite effective for solving P1 , it is natural to seek an adaptation of this algorithm for solving P2 that maintains A primal feasible *and integer* at each stage. The first step toward such an adaptation (the Rudimentary Primal Algorithm) is straightforward, and has apparently been rediscovered on several occasions (see, e.g., [1, 5, 6]).

4. THE RUDIMENTARY PRIMAL ALGORITHM (RPA)

Assume that the constant components of the initial A matrix are integers and the t_j are nonnegative integer variables.

As Ralph Gomory [4] has shown,equation (1) then implies the cut[†]

$$s = \left\lfloor a_{i0}/\lambda \right\rfloor + \sum_{j=1}^{n} \left\lfloor a_{ij}/\lambda \right\rfloor (-t_j) , \qquad (2)$$

where $\lambda > 0$ and s is a nonnegative integer variable. Thus $X = AT$ may be augmented to include (2) without altering the set of feasible integer solutions to P2 . Based on this, the RPA occurs by replacing instuction 3 of the PSA with the instruction 3A below.

3A. Select as the source equation for (2) any equation i such that $a_{is} > 0$ and $0 \leq \left\lfloor a_{i0}/a_{is} \right\rfloor \leq \theta_s$ (e.g., the equation v of instruction 3). Let $\lambda = a_{is}$, and designate (2) to be equation v of $X = AT$.

Designating (2) to be equation v is simply a symbolic device to specify the transformation of A into $\bar{A}$ by instruction 4 of the PSA (hence of the RPA). It is unnecessary to augment $X = AT$ by (2) if one is interested only in the values of the original variables x_i .

[†] $[u]$ denotes the greatest integer $\leq u$.

On the other hand, it is obvious that if equation (2) as determined by 3A were adjoined to $X = AT$, then it would qualify to be selected as equation v in instruction 3 of the PSA. Also, the coefficient a_{vs} bequeathed by instruction 3A is always 1 $\left([a_{is}/a_{is}]\right)$. These facts clearly assure that the successive A matrices determined by the RPA will be all integer and primal feasible. Unfortunately, however, there is no assurance that the RPA will converge to an optimal integer solution.

In a highly original paper [6], Richard D. Young showed how the RPA could be extended by the addition of a complex set of rules to produce a finite primal integer algorithm. Subsequently, drawing on certain ideas from [3], he was able to develop a much simpler set of rules whose justification, however, remained complicated. Relying still more heavily on [3], we now introduce an alternate framework which provides the simpler rules from a few elementary theorems, and in addition provides other rules that lead to a convergent algorithm.

5. THE SIMPLIFIED PRIMAL ALGORITHM

To produce a convergent primal integer algorithm, it suffices to[+]

(i) adjoin to $X = AT$ an additional primal feasible and all integer equation (call it r) whose coefficients a_{rj} satisfy certain properties in relation to the A_j,

(ii) select s in instruction 1 of the RPA by reference to the properties of equation r,

[+]One may stop and restart this procedure at finite intervals in the execution of the RPA if the number of digressions is finite. This can be assured for example by reliance on an intervening criterion of progress, such as an uninterrupted increase in the sum of the negative a_{0j} for $j \geq 1$, or in the value of a_{00} (R. D. Young's "transition cycles").

(iii) periodically select the source equation for (2) in instruction 3A of the RPA by reference to the size of the coefficients a_{is} (or other criteria to be introduced subsequently).

We develop the properties that we wish equation r to satisfy (in addition to being consistent with the other equations of $X = AT$), while simultaneously motivating and justifying the choice of s according to (ii). To this end, we introduce the following result:

Lemma:[†] Let $\bar{A}_j = A_j - kA_s$ for some scalar k. Then for each pair of indices i, r:

$$a_{rj}a_{is} < a_{rs}a_{ij} \quad (> a_{rs}a_{ij}\ ,\ = a_{rs}a_{ij})$$

if and only if

$$\bar{a}_{rj}a_{is} < a_{rs}\bar{a}_{ij} \quad (> a_{rs}\bar{a}_{ij}\ ,\ = a_{rs}\bar{a}_{ij})$$

Proof: By definition,

$$\bar{a}_{rj}a_{is} = (a_{rj} - ka_{rs})a_{is} = a_{rj}a_{is} - ka_{rs}a_{is}\ .$$

Also,

$$a_{rs}\bar{a}_{ij} = a_{rs}(a_{ij} - ka_{is}) = a_{rs}a_{ij} - ka_{rs}a_{is}\ .$$

Thus

$$\bar{a}_{rj}a_{is} - a_{rs}\bar{a}_{ij} = a_{rj}a_{is} - a_{rs}a_{ij}\ .$$

This proves the lemma.

[†]This lemma is essentially Lemma 2 of [3].

Note that the definition of $\bar{A}_j$ in the foregoing lemma accords with the definition of $\bar{A}_j$ for $j \neq s$ in instruction 4 of the RPA. Using this same definition, we now extend the result of the lemma to a lexicographic relationship[†] between vectors.

Theorem 1:

$$\text{(H1)} \quad a_{rj}A_s \overset{\ell}{<} a_{rs}A_j \quad (\overset{\ell}{>} a_{rs}A_j \ , = a_{rs}A_j)$$

if and only if

$$\text{(H2)} \quad \bar{a}_{rj}A_s \overset{\ell}{<} a_{rs}\bar{A}_j \quad (\overset{\ell}{>} a_{rs}\bar{A}_j \ , = a_{rs}\bar{A}_j)$$

Proof: Let $p = \text{Min}(i : \bar{a}_{rj}a_{is} \neq a_{rs}\bar{a}_{ij})$ and $q = \text{Min}(i : a_{rj}a_{is} \neq a_{rs}a_{ij})$. By the lemma, $p = q$ and Theorem 1 follows immediately.

Theorem 1 gives direct access to the properties we desire the reference equation r to possess. Indeed, we will want equation r to be created so that s may be selected to satisfy (H1) of Theorem 1. The power of Theorem 1 is that it implies, for s so selected, that equation r will still satisfy the same properties relative to the new matrix $\bar{A}$. In addition, when the appropriate choice rules are implemented, the theorem assures that a form of lexicographic progress will occur in passing from A to $\bar{A}$.

To make the foregoing precise, define

$$A_j^* = A_j / a_{rj}$$

[†]A vector A_h is defined to be lexicographically smaller than a vector A_k (symbolized $A_h \overset{\ell}{<} A_k$ or $A_k \overset{\ell}{>} A_h$) if and only if the first nonzero component of $A_k - A_h$ is positive.

for those $j \geq 1$ such that $a_{rj} \neq 0$. Then we specify the choice of s in instruction 1 so that:

$$a_{rs} > 0 \quad \text{and} \quad A_s^* \overset{\ell}{<} A_j^* \quad \text{for all} \quad j \neq s \ (s, j \geq 1) \text{ such that } a_{rj} > 0 .$$

Note that if there exists a j such that $A_j \overset{\ell}{<} 0$ and $a_{rj} > 0$, then A_s exists and is lexicographically negative. (There is clearly no need to consider the possibility $A_j^* = A_k^*$ for $j \neq k$ since the initial A_j for $j \geq 1$ include the - I matrix, and hence begin and remain linearly independent.)

The properties that we require equation r to satisfy are then as follows:[†]

$$\text{(P1)} \quad A_j \overset{\ell}{<} 0 \Rightarrow a_{rj} > 0 ,$$

$$\text{(P2)} \quad a_{rj} < 0 \Rightarrow A_j^* \overset{\ell}{<} A_s^* .$$

It may be observed that any equation with all positive coefficients will automatically satisfy both (P1) and (P2). In particular it suffices to create equation r initially with $a_{rj} = 1$ for all $j \geq 1$ and a_{r0} equal to an upper bound for $\sum t_j$.[††]

An interesting consequence of (P1) is that if $a_{0s} \geq 0$ (equivalently, $a_{0s}^* \geq 0$), then the relation $A_s^* \overset{\ell}{<} A_j^*$ for $a_{rj} > 0$ implies $a_{0j} \geq 0$ for all $j \geq 1$, and hence A is dual feasible. But $a_{0s} < 0$ implies that any change in

---

[†] These properties, as developed in connection with the Pseudo Primal-Dual Integer Programming Algorithm, are approximately equivalent to R. D. Young's subsequent definition of an "arranged tableau" in [8]. Note that (P2) => (P1) if there exists a j such that $a_{rj} > 0$ and $A_j \overset{\ell}{<} 0$.

[††] Under the assumption that P2 is bounded, any all-integer linear form $\sum a_{rj} t_j$ can be maximized by the simplex method subject to $X = AT$ to determine an upper bound a_{r0}. It is implied by our results to follow that if the a_{rj} satisfy (P1) and (P2), and if $a_{r0} < 1$, then $X = A_0$ already provides an optimal solution to (P2).

A_0 must produce an (integer) increase in a_{00}. Thus, in a bounded problem, the number of changes in A_0 must be finite whether the algorithm itself is finite or not. Our next theorem summarizes the joint implications of Theorem 1 and the properties (P1) and (P2) we have required of equation r.

Theorem 2:[†] If (P1) and (P2) are satisfied for the matrix A, then (H1) of Theorem 1 is true, and (P1) and (P2) are also satisfied for $\bar{A}$. Moreover, $\bar{A}_j^* \overset{\ell}{>} A_s^*$ for all j such that $\bar{a}_{rj} > 0$ (in particular, $\bar{A}_{\bar{s}}^* \overset{\ell}{>} A_s^*$, where $\bar{s}$ is defined relative to $\bar{A}$ as s is to A).

Proof: The definition of A_s^* directly implies (H1) for those $j \neq s$ such that $a_{rj} > 0$, and (P2) implies (H1) for those j such that $a_{rj} < 0$. If $a_{rj} = 0$, (H1) follows from (P1) and the fact that $A_j \neq 0$ for all j. Thus, (H2) is true by Theorem 1 and $\bar{A}_j \overset{\ell}{<} 0 \Rightarrow \bar{a}_{rj} > 0$. Hence (P1) is satisfied for $\bar{A}$. Also, dividing (H2) through by $\bar{a}_{rj}a_{rs}$ gives $\bar{A}_j^* \overset{\ell}{>} A_s^*$ when $\bar{a}_{rj} > 0$ and $A_s^* \overset{\ell}{>} \bar{A}_j^*$ when $\bar{a}_{rj} < 0$, $j \neq s$. The former proves the last assertion of the theorem, and the latter in conjunction with $\bar{A}_{\bar{s}}^* \overset{\ell}{>} A_s^*$ proves (P2) holds in $\bar{A}$ if $j \neq s$. But (P2) also holds if $j = s$ since $\bar{A}_s^* = A_s^*$. This completes the proof.

Having now established the form of equation r and the definition of the index s, it remains to specify the choice of the source equation for equation (2) in instruction 3A. The following result, in conjunction with Theorem 2, lays the foundation for this choice.

[†]Theorem 2, from results of [3], has a close resemblance to some of the results of Young's original primal algorithm [6], and very probably is collectively implied by them under appropriate reformulation.

Theorem 3:[†] If $\lambda = a_{is} > 0$ and equation (2) is designated to be equation v in instruction 4 of the RPA, then

$$\bar{a}_{is} = -a_{is} \quad \text{and}$$

$$a_{is} > \bar{a}_{ij} \geq 0 \quad \text{for} \quad j \neq s .$$

Proof: Since $a_{vs} = 1$, $\bar{a}_{is} = -a_{is}$. Also, for $j \neq s$, $\bar{a}_{ij} = a_{ij} - [a_{ij}/a_{is}]a_{is}$. The theorem follows at once from the fact that $u \geq [u] > u - 1$ for all u.

Note that if $a_{is} > a_{i0}$[††], then Theorem 3 implies that one may repeatedly select equation i as source equation for (2) in instruction 3A and thereby eventually assure $a_{is} \leq a_{i0}$, unless, of course, A becomes dual feasible in the interim.

By reference to this fact, R. D. Young then gives the following prescription for the selection of the source equation: use any rule that assures, for each $i \geq 1$ (including $i = r$), $a_{is} \leq a_{i0}$ will occur at finite intervals. Theorem 3 provides a ready mechanism for implementing this prescription, as indicated by our foregoing remarks.

We will here give some alternate choice rules that also produce a convergent primal algorithm and are easily justified within the framework of our present development. The first rule is slightly more flexible then the one given above.[†††]

[†] Theorem 3 was first introduced in the context of a primal algorithm by Young in [6], and in the context of a dual algorithm at about the same time by the author in [2].

[††] Since $a_{i0}/a_{is} \neq 0$, equation i is a permissible source equation in instruction 3A.

[†††] It is, however, close in spirit to Young's justification of his rule.

Rule 1: Make any choice that assures $a_{rs} \leq a_{r0}$ at finite intervals, and periodically reduces a^*_{is} for the least $i \geq 1$ such that $a^*_{is} > a_{i0}$.

We note that, since $a_{rs} \geq 1$ (by (P1)), it follows that $a_{is} \geq a^*_{is}$ for $a_{is} \geq 0$, and hence $(a^*_{is} > a_i) \Rightarrow (a_{is} > a_{i0})$ (equivalently, $(a_{is} \leq a_{i0}) \Rightarrow (a^*_{is} \leq a_{i0})$). Consequently, Rule 1 is meaningful and can be implemented by repeatedly selecting equation i as the source equation until a^*_{is} is decreased, unless A becomes dual feasible and the algorithm terminates first. We prove that this rule provides a finite algorithm as follows.

Justification of Rule 1: As observed earlier, A_0 can be changed only a finite number of times. Hence, for each i, there exists a finite constant U_i such that $a_{i0} \leq U_i$ for all values assumed by a_{i0}. Assume that Rule 1 is not finite. Then there is an infinite set T of A matrices in which $a_{rs} \leq a_{r0} \leq U_r$. Since A^*_s is lexicographically strictly increasing and $a_{0s} < 0$ it follows that a^*_{0s} can only assume a finite number of values in T and hence must eventually become constant (both in T and outside of T). Applying this argument to successive components of A^*_s, at least one of which must be unbounded, there exists an index $q \geq 1$ such that for all A matrices after an initial finite number (call the infinite remaining set of matrices S), $a^*_{qs} > U_q \geq a_{q0}$ and a^*_{is} is nondecreasing for all $i \leq q$. But at some point in S, Rule 1 will reduce a^*_{is} for some $i \leq q$ such that $a^*_{is} > a_{i0}$, which is impossible. This completes the justification.[+]

[+] It is clear from this proof that a_{r0} and a_{i0} may alternately be replaced by U_r and U_i in the specification of Rule 1. Also, it is unnecessary to require a^*_{is} to be decreased unless it has been nondecreasing for some finite duration.

The second rule we give has a somewhat different character than the first, drawing on additional results underlying the Pseudo Primal-Dual Algorithm. The Pseudo Primal-Dual Algorithm extablishes dual feasibility[†] by a strict increase in the first nonzero component of A_s at each step, thus providing a more evident push toward convergence than the successive lexicographic increases in A_s^*. In the interest of achieving comparable advances toward convergence with a primal algorithm, one is tempted to select a source equation whenever possible that will result in $\bar{a}_{0\bar{s}} > a_{0s}$. The somewhat surprising fact is that such a choice will indeed produce a finite algorithm, as we now show.

Rule 2: At finite intervals: select as source equation (if possible) one that will result in $\bar{a}_{0\bar{s}} > a_{0s}$, and continue the selection of source equations by this criterion until there are none that satisfy it.

As a basis for establishing the validity of this rule, we introduce -[††]

Theorem 4: For A and $\bar{A}$ satisfying conditions (P1) and (P2), and $a_{0s} < 0$:

$$(\bar{a}_{r\bar{s}} < a_{rs}) \Rightarrow (\bar{a}_{0\bar{s}} > a_{0s}).$$

Proof: By Theorem 2, (H1) is satisfied, hence by Theorem 1 $a_{0s}\bar{a}_{r\bar{s}} \leq a_{rs}\bar{a}_{0\bar{s}}$. Since $a_{0s} < 0$ and $a_{rs} > 0$, we have $\bar{a}_{0\bar{s}}/a_{0s} \leq \bar{a}_{r\bar{s}}/a_{rs} < 1$, hence $\bar{a}_{0\bar{s}} > a_{0s}$.

[†] A dual feasible matrix is driven dual infeasible in such a way that equation r occurs naturally in $X = AT$. Then dual feasibility is restored by continued selection of equation r as the source equation, and the net progress in A_0 between two consecutive instances of dual feasibility is at least as great as that produced by pivoting with the dual simplex method.

[††] This result abstracts a portion of Theorem 2 of [3]. Part of the remainder of the theorem is developed in the justification for Rule 2.

The completed proof that Rule 2 yields a finite primal algorithm is as follows:

<u>Justification of Rule 2:</u> Assume that the rule is not finite. By Theorem 3, if equation r is selected as the source equation, then $\bar{a}_{rs}^{-} < a_{rs}$ and hence by Theorem 4 $\bar{a}_{0s}^{-} > a_{0s}$. Thus, Rule 2 assures $a_{rs} \leq a_{r0}$ foran infinite number of A matrices. Identify q and S as in the justification for Rule 1. If q were selected as source equation in S (it is always eligible), we have $\bar{a}_{qs}^{-} < a_{qs}$ (by Theorem 3), and this in conjunction with $\bar{a}_{qs}^{*-} \geq a_{qs}^{*}$ implies $a_{rs} > \bar{a}_{rs}^{-}$. Then by Theorem 4, for every A matrix in S there is a source equation available (namely q) that will yield $\bar{a}_{0s}^{-} > a_{0s}$. Rule 2 will thus produce an infinite number of consecutive integer increases in a_{0s}, which is impossible.[†]

Until now we have assumed that the algorithm terminates only when A becomes dual feasible. A basis for eliminating this assumption, and hence for increasing the effectiveness of the choice rules, is embodied in the following theorem.[††]

[†] It may be noted that Rule 2 remains valid by this proof when the condition $\bar{a}_{rs}^{-} < a_{rs}$ replaces $\bar{a}_{0s}^{-} > a_{0s}$. An argument mirroring the proof of Theorem 4 also shows that selecting equation i as source equation will result in $\bar{a}_{rs}^{-} < a_{rs}$ whenever $\bar{a}_{is}^{-*} \geq a_{is}^{*}$. Other rules closely related (but not equivalent) to the foregoing ones can immediately be inferred from this: e.g., enforce $a_{rs} \leq a_{r0}$ at finite intervals and periodically select any source equation i such that $a_{is}^{*} > a_{i0}$ and $\bar{a}_{is}^{-*} \geq a_{is}^{*}$, continuing the selection of i by this criterion as long as possible.

[††] This theorem can alternately be justified by the observation by R. D. Young that an appropriate multiple of row r provides a feasible solution to the dual of P1 (defined for the current A matrix). In this connection, Ben-Israel and Charnes [1] anticipate the present use of this result by advocating the solution of an auxiliary problem to prescribe termination before A is dual feasible.

Theorem 5: Assume that A is dual infeasible and satisfies (P1) and (P2). Then the optimal value of x_0 must satisfy

$$x_0 \leq a_{00} + \left[-a_{r0}a^*_{0s}\right] .$$

Proof: Let P2' be the problem obtained from P2 by introducing a new variable t_w, and replacing each equation of P2 by

$$x_i = a_{i0} + \sum_{j=1}^{n} a_{ij}(-t_j) + a_{iw}(-t_w) , \quad i = 0, 1, \ldots, m + n + 1 ,$$

(where, e.g., $r = m + n + 1$), and adjoining the two additional equations

$$x_{m+n+2} = -(-t_w) , \quad x_{m+n+3} = -t_w .$$

These last equations assure that every optimal solution P2' is also optimal for P2, disregarding t_w, x_{m+n+2} and x_{m+n+3} (which must all equal 0). Let $a_{iw} = -a_{is}/a_{0s}^{+}$ for $i = 0, 1, \ldots, m + n + 1$. Then clearly w satisfies the requirements for s in P2'. Suppose equation r is selected as the source equation in P2', with a_{rw} taking the place of a_{rs} (whether or not $\frac{a_{r0}}{a_{rw}} \leq \theta_w$).

It follows from Theorems 3 and 4 that $\bar{a}_{0\bar{w}} > a_{0w}$, where $\bar{w}$ denotes the index that corresponds to $\bar{s}$ in P2'. But $a_{0w} = -1$, hence $\bar{a}_{0\bar{w}} = 0$ (it must be an integer) and $\bar{A}$ is dual feasible in P2' (though possibly not primal feasible). Consequently, $\bar{a}_{00}$ (in P2') provides an upper bound for x_0. But from instruction 4 of the RPA, with w replacing s, we have $\bar{a}_{00} = a_{00} + \left[-a_{r0}a^*_{0s}\right]$, since $a_{0w} = -1$ and

$$a_{rw} = -\frac{a_{rs}}{a_{0s}} = -\frac{1}{a^*_{0s}} .^{\dagger}$$

It is immediate from Theorem 5 that $X = A_0$ provides optimal solution to P2 whenever $a^*_{0s} > -\frac{1}{a_{r0}}$. This fact not only permits earlier termination of the algorithm, but also gives rise to an additional choice rule.

Suppose that $X = AT$ is augmented by the equation

$$x_u = a_{u0} + \sum_{j=1}^{n} a_{uj}(-t_j)$$

where $a_{uj} = -a_{0j}$ for $j \geq 1$ and $-a_{u0}$ is a nonpositive lower bound for $\sum a_{0j}t_j$. Then we have

Rule 3: Replace $a_{rs} \leq a_{r0}$ by $a_{us} \leq a_{u0}$ in Rule 1, and terminate when $a^*_{0s} > \frac{1}{a_{r0}}$ (if A is not dual feasible).††

Justification for Rule 3: The justification is the same as for Rule 1 if $a_{rs} \leq C$ occurs periodically for some finite constant C. Otherwise, if the algorithm doesn't converge, $a_{rs} > U_u \cdot U_r \geq a_{u0}a_{r0}$ for all A matrices except an initial finite number. In some of the subsequent matrices $a_u \geq a_{us} = -a_{0s}$ and hence $a^*_{0s} > -\frac{1}{a_{r0}}$, contrary to the nonconvergence assumption.

---

† More generally in defining P2' let $a_{iw} = -ka_{is}/a_{0s}$ for $k > 0$. Then selecting r as a source equation yields $\bar{a}_{0j} = a_{0j} + k[\rho_j/k]$ for $j = 0, 1, \ldots, n$ where $\rho_j = -a_{rj}a^*_{0s}$. Thus a more restrictive upper bound on x_0 is obtained by selecting $k > 0$ to minimize $a_{00} + k[\rho_0/k]$ subject to $k[\rho_j/k] \geq a_{0j}$ (to assure $\bar{a}_{0j} \geq 0$), $j = 0, 1, \ldots, n$. Note that a_{0s}/k must be an integer to satisfy this latter inequality for $j = s$.

†† Since a^*_{0s} is nondecreasing, $a_{rs} \leq a_{r0}$ implies $a_{rs} \leq a_{u0}$ for an appropriately large value of a_{u0}, and hence Rule 3 includes Rule 1 as a special instance.

REFERENCES

[1] Ben-Israel, A. and A. Charnes, "On Some Problems of Diophantine Programming," Cahiers du Centre L'Etudes de Recherche Operationnelle, Brussels, (1962).

[2] Glover, Fred, "A Bound Escalation Method for the Solution of Integer Linear Programs," Cahiers du Centre L'Etudes de Recherche Operationnelle, Vol. 6, Brussels, (1964).

[3] Glover, Fred, "An Extension of the Bound Escalation Method for Integer Programming: A Pseudo Primal-Dual Algorithm of the Gomory All-Integer Variety," Management Sciences Research Report No. 49, Carnegie Institute of Technology, (July 1965).

[4] Gomory, Ralph E., "All-Integer Integer Programming Algorithm," Industrial Scheduling, J. F. Muth and G. L. Thompson, eds., Prentice Hall, (1963).

[5] Gonzalez-Zubieta, Romulo H., "Fundamental Investigations in Methods of Operations Research," Technical Report No. 16, Massachusetts Institute of Technology, (June 1965).

[6] Young, R. D., "A Primal (All-Integer) Integer Programming Algorithm: Antecedents, Description, Proof of Finiteness, Exemplification," Working Paper no. 52, Stanford University, (December 1964).

[7] Young, R. D., "A Primal (All-Integer) Integer Programming Algorithm," Rice University, (12 June 1965).

[8] Young, R. D., "A Simplified Primal (All-Integer) Integer Programming Algorithm," Rice University, (September 1966).

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)	2a. REPORT SECURITY CLASSIFICATION
The University of California	Unclassified
	2b. GROUP

3. REPORT TITLE

A NEW FOUNDATION FOR A SIMPLIFIED PRIMAL INTEGER PROGRAMMING ALGORITHM

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Research Report

5. AUTHOR(S) (Last name, first name, initial)

Glover, Fred

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
November 1966	18	8

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
Nonr-222(83)	
b. PROJECT NO	ORC 66-39
NR 047 033	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d	

10. AVAILABILITY/LIMITATION NOTICES

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
Also supported by the National Science Foundation, Grant GP-4593	Office of Naval Research Mathematical Science Division

13. ABSTRACT

Following the approach underlying the Pseudo Primal-Dual Integer Programming Algorithm, new foundation for the Young Simplified Primal Integer Programming Algorithm is given. Simplifications in conception and presentation are developed to produce a primal integer algorithm that is particularly easy to justify and implement. In addition, new choice rules are prescribed which guarantee finite convergence, and a criterion of optimality is introduced that permits the algorithm to terminate before dual feasibility is achieved.

DD FORM 1 JAN 64 1473

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Integer Programming Primal Algorithm All-Integer						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year, or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through _____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS), (S), (C), or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.